Tries

↙ ☆ Store and search strings efficiently

Tries: Example

$\sigma \dots 25$

$a, \dots, z$

empty string

ε

$C\alpha$
car
$cars$

a	b	ⓒ	...	y	z	final:
						false

False indicates this character is not end.

a	b	c	...	y	z	final:
						false

a	...	r	...	t	...	final:
						false

a	b	...	s	...	z	final:
						~~false~~

a	b	c	...	y	z	final:
						true

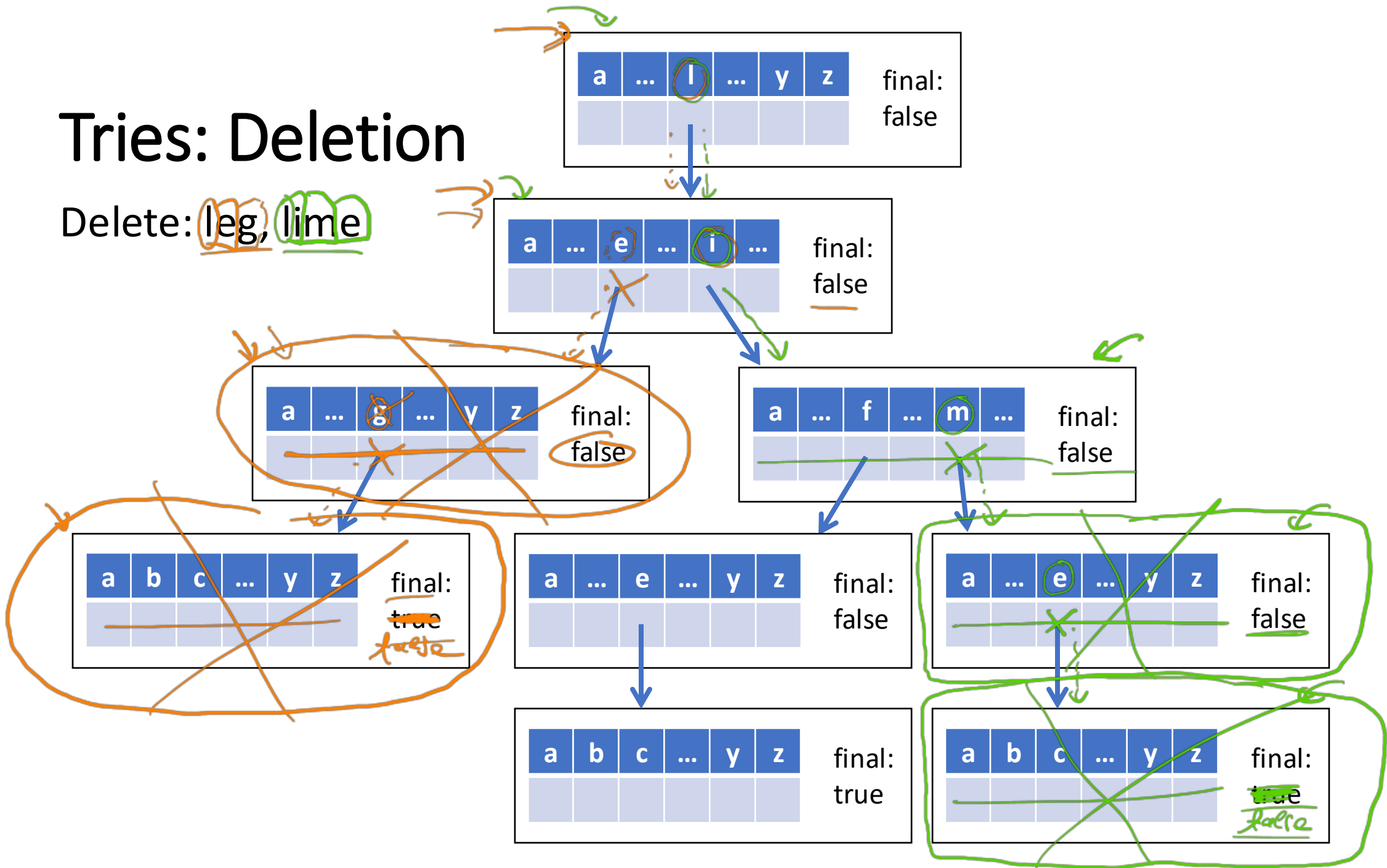| a | b | c | ... | y | z | final: |
|---|---|---|---|---|---|
| | | | | | | ~~false~~ |

cat
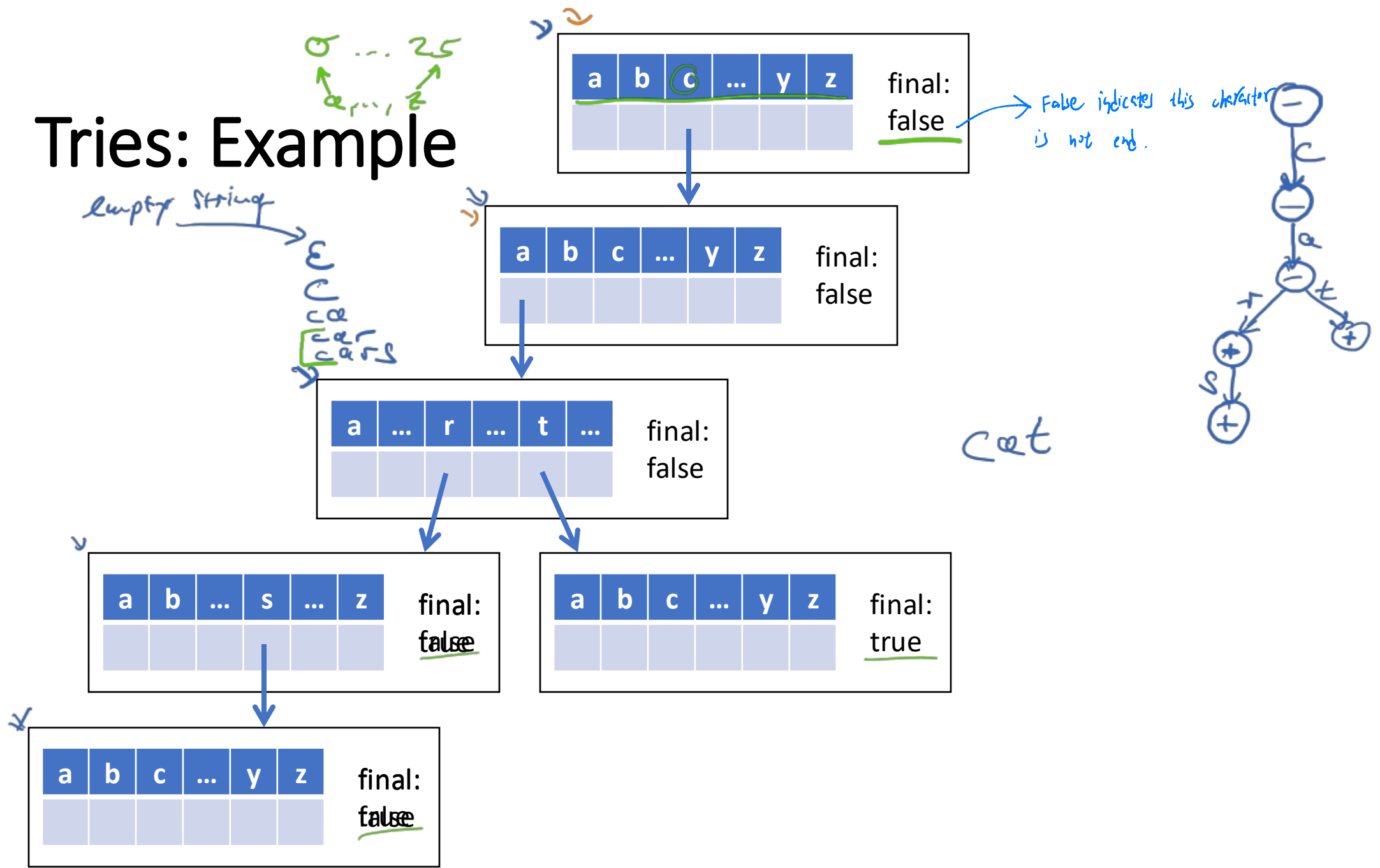
Tries: Insertion

Insert: life, lime, leg
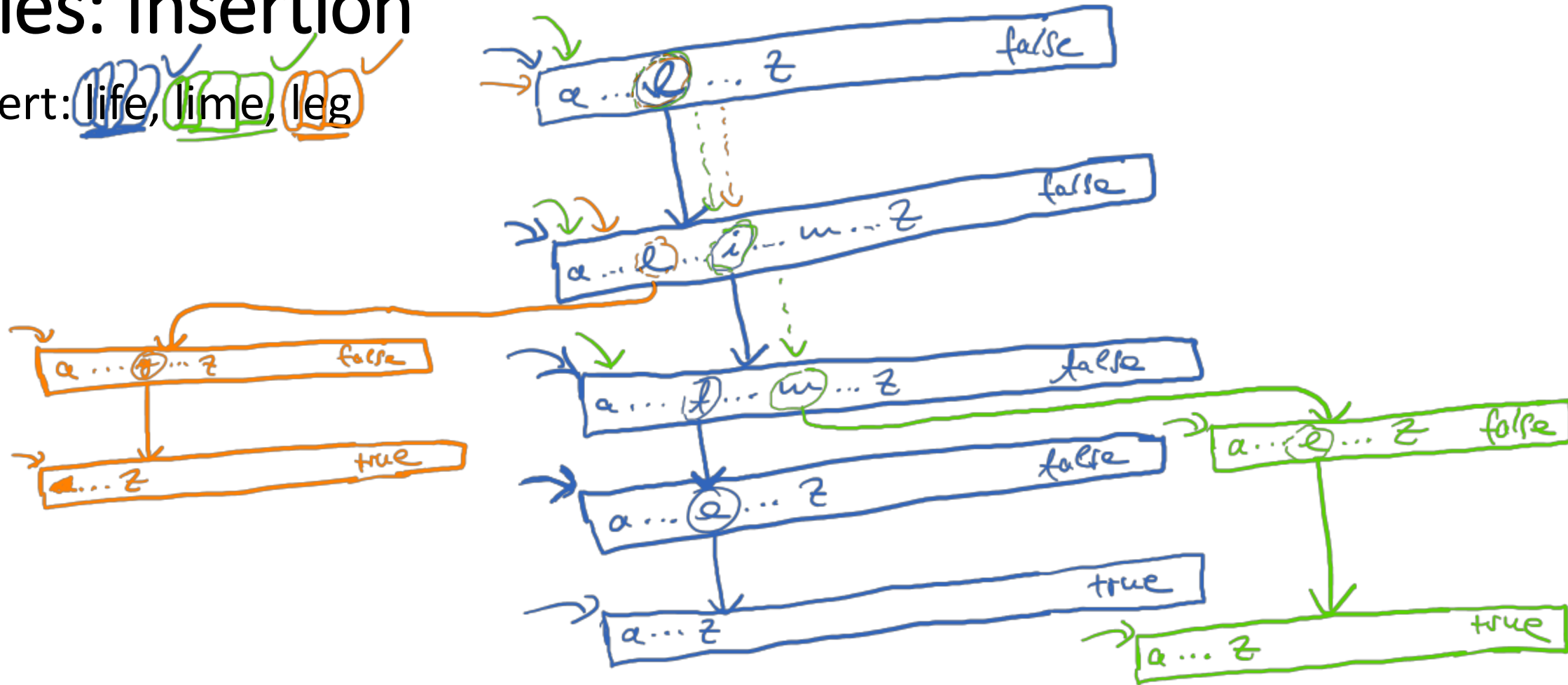
Tries: Deletion

Delete: leg, lime

Complexities

N: # strings in trie

L: length of string (# chars)

- lookup: $O(L + \cancel{N})$

- insert: $O(L + \cancel{N})$

- delete: $O(\cancel{2} \cdot (L + \cancel{N}))$